PROVA SCRITTA DI FONDAMENTI DI INFORMATICA L A

4 DICEMBRE 2003

Prof. Michela Milano

COMPITO A

Esercizio 1 (punti 5)

Si scriva una funzione ricorsiva `int doppio(int a, int n);` che calcoli il seguente valore:

$$\sum_{i=1}^{n} 2*(a+i)$$

Esercizio 2 (punti 8)

Dato il seguente programma C:

```
#include <stdio.h>
#define Costante 7

void cambia(int V1[], int V2[], int V3[], int n);
void stampa(int vet[],int j);


void main()
{
int A[5]={6,12,20,25,32};
int B[3]={2,4,7};
int C[5]={1,2,3,4,5};
int p,q;
 p = 3;
 stampa(C,p);
 cambia(A,B,C,Costante-2);
 q = ++p;
 q++;
 stampa(C,q);
 printf("%d ",q);
 }

void cambia(int V1[], int V2[], int V3[], int n)
{int i;
 for(i=0;i<n;i++)
   if (V1[i]%V2[1] == 0) V3[i] = V1[i];
}

void stampa(int vet[], int k)
{int i;
 for(i=0;i<k;i++)
  printf("%d ",vet[i]);
 printf("\n");
}
```

Cosa viene stampato dal programma? La variabile ***i*** definita all'interno della procedura **`cambia`** e' visibile nel **`main`**? Qual e il suo tempo di vita? Le risposte devono essere opportunamente motivate.

Esercizio 3 (punti 6)

Dato una struttura **`voto`**

```
struct voto {char[20] studente;
             int matricola;
             int punteggio;}
```

Si scriva una procedura che prende in ingresso un vettore V di struct voto, una matricola e restituisca in uscita la somma dei punteggi per quella matricola

```
void somma(struct voto V[], int Matr, int* somma_punt);
```

Si scriva l'interfaccia della funzione che restituisce lo stesso valore.

Si scriva un possibile main che richiami la procedura e stampi la somma ottenuta.

Esercizio 4 (punti 6)

Data la seguente funzione ricorsiva:

```
int p(int a, int b)
{     if (a > b) return 1;
      else return a * p(a+1,b-1);
}
```

si dica qual è il valore fornito dalla funzione e si disegni la sequenza di record di attivazione nel caso in cui la funzione venga invocata con i seguenti parametri attuali:

`p(3,6)`

Esercizio 5 (punti 7)

- Cosa succede quando si dichiara un intero short int (che come ordine di grandezza va da $-32 * 10^3$ a $+32 * 10^3$) e poi si effettuano operazioni che danno un problema di overflow.?
- Cosa indica il nome di un vettore?
- Si possono assegnare due strutture S1 e S2 che hanno i medesimi campi tramite un assegnamento diretto del tipo S1 = S2? Perché?
- Cosa si intende per gerarchia di memoria?
- Quale è il ruolo di un file system?

PROVA SCRITTA DI FONDAMENTI DI INFORMATICA L A

4 DICEMBRE 2003

Prof. Michela Milano

COMPITO B

Esercizio 1 (punti 5)

Si scriva una funzione ricorsiva `int quattro(int n, int b);` che calcoli il seguente valore:

$$\sum_{i=1}^{n} 4*(b+i)$$

Esercizio 2 (punti 8)

Dato il seguente programma C:

```
#include <stdio.h>
#define DIM 7

int X(int V[], int p)
{     int k; int r;
      r = p;
      for (k=0; k<DIM; k++) {
            if (V[k] > 0)
                  r = r + V[k];
            else
                  V[k] = -V[k];
      }
      return r;
}

void main()
{
      int i; int A[DIM]={0, 2, -2, 4, -4, 6, -6};
      for (i=DIM-1; i>=0; i--)
            A[i] = A[0] - A[i];
      for (i=0; i<DIM; i++)
            printf("%d  ", A[i]);

      printf("\n");
      printf("%d", X(A, 4));
      printf("\n");

      for (i=0; i<DIM; i++)
            printf("%d  ", A[i]);
}
```

Cosa viene stampato dal programma? La variabile ***k*** definita all'interno della funzione **`X`** e' visibile nel **`main`**? Qual e il suo tempo di vita? Le risposte devono essere opportunamente motivate.

Esercizio 3 (punti 6)

Dato una struttura **`punti`**

```
struct punti {char[20] nome;
              int numero_patente;
              int punto;}
```

Si scriva una procedura che prende in ingresso un vettore P di struct punti, un numero patente e restituisca in uscita la somma dei punteggi rimossi da quella patente

```
void somma(struct punti P[], int Num_Pat, int* somma_punt);
```

Si scriva l'interfaccia della funzione che restituisce lo stesso valore.

Si scriva un possibile main che richiami la procedura e stampi la somma ottenuta.

Esercizio 4 (punti 6)

Data la seguente funzione ricorsiva:

```
int f(int a, int b)
{if (a>b)
      return (a-b)*4;
      else return f(a+2, b);
}
```

Si dica qual è il valore restituito dalla funzione e si disegnino i record di attivazione nel caso in cui la funzione sia chiamata con i seguenti parametri attuali **`f(0,5)`**.

Esercizio 5 (punti 7)

- Cosa succede quando si dichiara un intero short int (che come ordine di grandezza va da $-32 * 10^3$ a $+32 * 10^3$) e poi si effettua una somma che esce dall'intervallo di definizione? Quale risultato ci attendiamo?
- Si possono assegnare due array con lo stesso numero di elementi dello stesso tipo tramite un assegnamento del tipo V1 = V2? Perché?
- Cosa accade quando si modifica una struttura all'interno di una procedura?
- Cosa si intende per unità aritmetico-logica?
- Quale è il ruolo di un sistema operativo?

COMPITO A

Esercizio 1

```
int doppio(int a, int n)
{ if (n==1) {return 2*(a+1);}
    else return 2*(a+n) + doppio(a,n-1);
}
```

Esercizio 2

La prima esecuzione della procedura **stampa** scrive in output i primi 3 elementi del vettore C. L'output sarà quindi:
1 2 3

La procedura **cambia** pone in V3[i] il valore di V1[i] per ogni V1[i] divisibile per V2[1]. L'indice i varia tra 0 e 4. Quindi, dopo l'esecuzione della procedura, il vettore V3 sarà:
V3 = [1, 12, 20, 4, 32].
I cambiamenti del vettore V3 saranno visibili anche nel **main** perché i vettori vengono passati come parametri per riferimento.

q viene dapprima assegnato a p+1 e poi incrementato di 1, quindi varrà 5.

Le ultime due istruzione scrivono in output:
1 12 20 4 32
5

Riassumendo l'output prodotto sarà:
1 2 3
1 12 20 4 32
5

La variabile **i** definita all'interno della procedura **cambia** non è visibile dal **main**.
Il suo tempo di vita è l'intera esecuzione della procedura stessa.

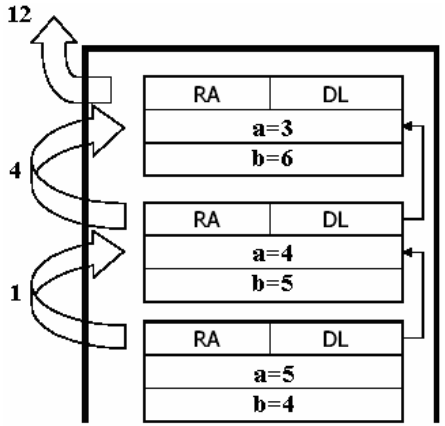
Esercizio 3

```
#include <stdio.h>
#define N 10

struct voto {char studente[20];
             int matricola;
               int punteggio;
               };
void somma(struct voto V[], int Matr, int* somma_punt)
{int i;
 *somma_punt=0;
 for(i=0;i<N;i++)
    if (V[i].matricola==Matr) *somma_punt+=V[i].punteggio;
}

// FUNZIONE : int somma(struct voto V[], int Matr);

void main()
{int i, soma_voti;
 struct voto Vett[N];
 for (i=0;i<N;i++)
    {scanf("%s",Vett[i].studente);
     scanf("%d",&Vett[i].matricola);
     scanf("%d",&Vett[i].punteggio);
    }
 somma(Vett, 20, &somma_voti);
 printf("%d",somma_voti);
}
```

Esercizio 4

Il valore fornito dalla funzione è 12.
Invocando p(3,6) si ottiene la sequenza di chiamate:
p(3,6) → p(4,5) → p(5,4).

Il record di attivazione è:



Esercizio 5

- Cosa succede quando si dichiara un intero short int (che come ordine di grandezza va da $-32 * 10^3$ a $+32 * 10^3$) e poi si effettuano operazioni che danno un problema di overflow?

Si ha un problema di overflow quando il risultato di operazione supera il limite massimo per il tipo di dato utilizzato. Nel caso dello short int, il limite è 32k, quindi qualsiasi operazione che fornisca come risultato un numero maggiore di 32k produce un overflow. Il massimo numero trattabile è composto, in notazione binaria, da 16 bit tutti uguali ad 1. Sommando 1 a tale numero, si dovrebbe ottenere un numero di 17 cifre in cui solo quella più significativa uguale ad 1 e tutte le altre uguali a 0. Ma dato che lo short int tratta solo 16 bit, il riporto sul diciassettesimo bit è perso e il risultato sarà un numero binario di 16 cifre tutte uguali a 0: in notazione complemento a 2, tale numero corrisponde al numero decimale (-32k+1), che è il limite minimo per il tipo short int.

- Cosa indica il nome di un vettore?

Il nome di un vettore indica l'indirizzo iniziale della locazione di memoria allocata al vettore stesso.

- Si possono assegnare due strutture S1 e S2 che hanno i medesimi campi tramite un assegnamento diretto del tipo S1 = S2? Perché?

Sì, perché il nome della struttura rappresenta la struttura nel suo complesso e non un puntatore alla locazione di memoria in cui è contenuta la struttura stessa.

- Cosa si intende per gerarchia di memoria?

Per gerarchia di memoria si intende l'organizzazione delle memorie presenti in un elaboratore secondo la loro velocità di accesso e la loro capacità di memorizzazione. A livello più alto della gerarchia si sono i registri del processore, a velocità d'accesso elettronica ma a capacità molto limitata (32 o 64 bit). Scendendo nella gerarchia troviamo le memorie cache del processore, utilizzate per memorizzare all'interno del processore (e quindi a velocità d'accesso elettronica) le porzioni di memoria centrale correntemente utilizzate. La loro capacità è limitata dal loro elevato costo e tipicamente è dell'ordine del Mb. Al successivo livello di gerarchia troviamo le memorie centrali (RAM), molto più capienti (centinaia di Mb) ma con velocità di accesso dell'ordine delle decine di ns. Al livello più basso troviamo le memorie di massa, con capacità molto elevate (decine o centinaia di Gb), ma con velocità di accesso estremamente lente. Per memorie di massa ad accesso casuale il tempo di accesso è dell'ordine dei ms, mentre per memorie ad accesso sequenziale il tempo di accesso dipende dalla posizione corrente di lettura.

- Quale è il ruolo di un file system?

Il ruolo del file system è l'organizzazione e la memorizzazione su disco dei file.
Il file system può essere suddiviso in cartelle innestate gerarchicamente.

COMPITO B

Esercizio 1

```
int quattro(int n, int b)
{ if (n==1) {return 4*(b+1);}
    else return 4*(b+n) + quattro(n-1,b);
}
```

Esercizio 2

All'inizio il vettore **A** contiene **[0, 2, -2, 4 , -4, 6, -6]**.
Successivamente viene modificato ogni elemento con la sottrazione tra **0** e l'elemento stesso. In pratica viene cambiato il segno ad ogni elemento di **A**.
Successivamente viene fornito in output tutto il vettore **A** modificato:
0 -2 2 -4 4 -6 6

Viene quindi calcolato il risultato della funzione **X(A, 4)**. Tale funzione somma ad una variabile **r**, che inizialmente vale **p**, tutti gli elementi di **V** maggiori di 0. Inverte il segno a tutti quelli minori di 0.
Il risultato della funzione chiamata con i parametri **A** e **4** è **16**.
Tale risultato viene anche fornito in output, e viene poi scritto il nuovo vettore **A**. Le modifiche fatte dalla funzione saranno visibili anche nel **main** perché i vettori vengono passati come parametri per riferimento.
L'output è:
16
0 2 2 4 4 6 6

Riassumendo l'output prodotto sarà:
0 -2 2 -4 4 -6 6
16
0 2 2 4 4 6 6

La variabile **k** definita all'interno della funzione **X** non è visibile dal **main**.
Il suo tempo di vita è l'intera esecuzione della funzione stessa.

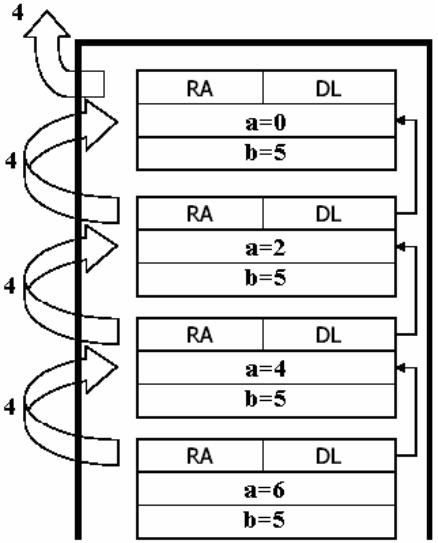
Esercizio 3

```
#include <stdio.h>
#define N 10

struct punti {char nome[20];
              int numero_patente;
              int punto;
             };

void somma(struct punti P[], int Num_Pat, int* somma_punt)
{int i;
 *somma_punt=0;
 for(i=0;i<N;i++)
    if (P[i].numero_patente==Num_Pat)
       *somma_punt+=P[i].punto;
}

// FUNZIONE : int somma(struct punti P[], int Num_Pat);

void main()
{int i, somma_punti;
 struct punti Vett[N];
 for (i=0;i<N;i++)
    {scanf("%s",Vett[i].nome);
     scanf("%d",&Vett[i].numero_patente);
     scanf("%d",&Vett[i].punto);
    }
 somma(Vett, 20, &somma_punti);
 printf("%d",somma_punti);
}
```

Esercizio 4

Il valore fornito dalla funzione è 4.
Invocando f(0,5) si ottiene la sequenza di chiamate:
f(0,5) → f(2,5) → f(4,5) → f(6,5).

Il record di attivazione è:



Esercizio 5

- Cosa succede quando si dichiara un intero short int (che come ordine di grandezza va da $-32 * 10^3$ a $+32 * 10^3$) e poi si effettua una somma che esce dall'intervallo di definizione? Quale risultato ci attendiamo?

Se si esegue un'operazione di somma che esce dall'intervallo di definizione si ha un problema di overflow. Esso si presenta quando il risultato di operazione supera il limite massimo per il tipo di dato utilizzato. Nel caso dello short int, il limite è 32k, quindi qualsiasi operazione che fornisca come risultato un numero maggiore di 32k produce un overflow. Il massimo numero trattabile è composto, in notazione binaria, da 16 bit tutti uguali ad 1. Sommando 1 a tale numero, si dovrebbe ottenere un numero di 17 cifre in cui solo quella più significativa uguale ad 1 e tutte le altre uguali a 0. Ma dato che lo short int tratta solo 16 bit, il riporto sul diciassettesimo bit è perso e il risultato sarà un numero binario di 16 cifre tutte uguali a 0: in notazione complemento a 2, tale numero corrisponde al numero decimale (-32k+1), che è il limite minimo per il tipo short int. Sommando a+b, se si supera 32k, il risultato che ci attendiamo è quindi $(a + b) - 2^{16} = (a + b) - 64k$.

- Si possono assegnare due array con lo stesso numero di elementi dello stesso tipo tramite un assegnamento del tipo V1 = V2? Perché?

No, perché il nome di un vettore rappresenta un indirizzo di memoria, e non il vettore stesso. Occorre assegnare un elemento per volta V1[i]=V2[i] .

- Cosa accade quando si modifica una struttura all'interno di una procedura?

Per tutto il tempo di vita della procedura la struttura risulta modificata, ma ritornando al chiamante le modifiche vengono perse perché le strutture sono passate come parametri per copia alle procedure e alle funzioni.

- Cosa si intende per unità aritmetico-logica?

Per Unità Aritmetico-Logica (ALU) si intende un'unità, fisicamente posta all'interno del processore, in cui compito è quello di eseguire tutte le operazioni logiche ed aritmetiche elementari.

- Quale è il ruolo di un sistema operativo?

Il sistema operativo (S.O.) si pone tra utente e macchina fisica e fornisce un'astrazione (macchina virtuale) tramite la quale l'utente riesce ad interfacciarsi con la macchina utilizzando operazioni di più alto livello rispetto alle istruzioni in codice macchina. Il S.O. gestisce tutte le risorse fisiche, quali le memorie, il file system, le periferiche, ecc., ed interpreta i comandi di alto livello richiesti dall'utente trasformandoli in istruzioni più semplici eseguite poi dai componenti fisici sottostanti come sequenze di impulsi elettrici.